JOURNAL OF BASIC SCIENCE AND ENGINEERING

Optimised Disease Detection in Avocado Crop using Deep Learning Employing Vision Transformer

Velammal M	Rajeswari V	Atishri Darshine V	Dharnesh V	Theoder Harish T	Yasvanthkumar P
Assistant Professor	Professor	Final Year / CT	Final Year / CT	Final Year / CT	Final Year / CT
Department of CT	Department of CT	Department of CT	Department of CT	Department of CT	Department of CT
Karpagam College of Engineering,	Karpagam College of Engineering,	Karpagam College of Engineering,	Karpagam College of Engineering,	Karpagam College of Engineering,	Karpagam College of Engineering,
Coimbatore,India	Coimbatore,India	Coimbatore,India	Coimbatore,India	Coimbatore,India	Coimbatore,India
velammal.cst@gmail.com	rajeswari.cst@gmail.com	atishridarshine@gmail.com	dharaneshv31@gmail.com	theoderharish2001@gmail.com	yasvanthkumarprabu@gmail.com

Abstract

Fruit diseases have a major impact on both the quantity and quality of fruit produced. Because of its high nutritional content, one of the world's most significant food sources is avocado fruit. Avocado fruit diseases reduce crop yield and result in financial losses for growers. Diseases like scab, stem end rot, rat bite, seed moth, sun bloch, phytophthora, anthracnose, and cercospora spot can affect fruit. Thus, it is crucial to identify illnesses early in order to take preventative action and stop the spread of disease. Our new deep learning-based approach is presented in this paper for classifying avocado fruit diseases. To enhance the performance of the model, this paper starts with preprocessing the data using normalisation techniques. After that, it uses ResNet-50 for feature extraction, taking advantage of its deep architecture to identify complex patterns in pictures of avocado fruits. We use a vision transformer for classification, making use of its attention mechanism to reliably identify patterns of disease and classify images. It effectively tunes hyperparameters using Cat and Mouse-Based Optimisation (CMBO) to maximise model performance. The suggested model outperforms current models with a 99.98% classification accuracy, achieving impressive results. The goal of this research is to improve orchard management techniques and enable prompt intervention by developing reliable and accurate automated tools for detecting avocado fruit disease.

Keywords: Avocado Disease, Residual Network, Vision transformer, Cat and Mouse-Based optimization, Normalization.

1. Introduction

A crucial role for agriculture is played in cultivation and harvesting. Fruit quantity and quality are significantly impacted by plant and fruit diseases [1]. Uneven climatic conditions are having an impact on fruits, resulting in lower agricultural yield. This has an effect on the global agriculture economy. Furthermore, if the fruits have any kind of disease, the situation will only get worse. Here is where the goal of current agricultural structures and strategies is to recognise these diseases and keep the fruits from being impacted by them [2]. One of the most sought-after dietary sources in the world is avocado fruit. The loss of production yield and quality resulting from diseases and agroclimatic conditions is approximately 20% [3]. The avocado, or Persea Americana, is an important fruit that is indigenous to Mexico and Central America and grows practically everywhere in the world's

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

tropical as well as subtropical zones. Avocado fruits are extremely nutrient-dense, with high concentrations of vitamins, minerals, proteins, fibre, and unsaturated fatty acids all of which are beneficial to health [4]. The avocado fruit is highly prized for its role in the health and cosmetics industries as well as its high nutritional content. Early detection of fruit diseases is essential because they have an impact on the farming sector. The most common diseases that affect avocado fruit are Fruits and leaves of trees are affected by anthracnose (caused by Colletotrichum gloeosporioides), phytophthora (caused by phytophthora cinnamomi), and stem-end rot (caused by Dipladia sp.). Avocado tree tissues are susceptible to the common and annoying fungus known as "cercospora spot" [5]. Early disease detection enables control of the illness through the implementation of certain preventative measures [6]. Deep learning models have gained popularity recently for automatically identifying plant diseases [7].

Hailed as the "green gold," avocado has become a profitable commodity on the international market because of its rich nutritional content and plethora of culinary uses [8]. However, a number of diseases that impact fruit quality, yield, and overall plant health pose serious threats to its cultivation [9]. Among these are the well-known illnesses avocado sunblotch (caused by Avocado Sunblotch Viroid; ASBVd) and anthracnose (caused by Colletotrichum species). In addition to causing growers to suffer significant financial losses, these illnesses jeopardise the supply of premium avocados to consumers across the globe [10]. Due to a number of factors, identifying avocado fruit diseases is a difficult task for both researchers and growers. First of all, diseases frequently have subtle symptoms, making early detection and treatment challenging. Furthermore, the labor-intensive, time-consuming, and subjective nature of traditional disease identification techniques like visual inspection and manual sampling results in inconsistent diagnosis [11]. Furthermore, the various environmental circumstances that avocados are grown in make it more difficult to identify diseases because symptoms can differ based on soil type, climate, and cultural customs. Therefore, in order to enable prompt management strategies, there is an urgent need for sophisticated, automated systems that can accurately and efficiently classify avocado fruit diseases [12].

"Deep learning," a subset of artificial intelligence, is motivated by the structure and operations of the human brain, which has become a potent instrument for agricultural disease diagnosis and treatment [13]. Deep learning models have shown impressive abilities in image recognition and classification tasks by using intricate neural networks to learn complex patterns and features from large datasets. Deep learning has enormous potential to completely change how diseases are found and tracked in orchards when it comes to classifying avocado fruit diseases [14]. Deep learning algorithms can accurately identify subtle disease symptoms from high-resolution images of avocado fruits, allowing for early intervention and reducing yield losses. Furthermore, deep learning systems' adaptability and scalability make them ideal for implementation in a variety of agricultural contexts, providing a long-term remedy to the problems caused by avocado diseases [15].

1.1. Motivation

The classification of avocado fruit diseases is driven by the urgent need to protect avocado farming, a sector of the economy that is heavily threatened by a number of diseases. Due to its high nutritional content and wide range of culinary applications, avocados have become more and more popular worldwide, leading to a rise in their cultivation. Diseases like anthracnose, sunblotch, and other fungal infections, however, pose a serious threat to this

growth because they not only reduce fruit quality and yield but also result in large financial losses for growers. In order to properly manage these illnesses, it is imperative that they are quickly and accurately identified lessen their negative effects on avocado production. The labor-intensive, subjective, and error-prone nature of traditional disease diagnosis techniques emphasises the need for sophisticated, automated systems. Through the application of deep learning techniques, scientists hope to create reliable models that can recognise and categorise diseases affecting avocado fruit from photos, allowing for early intervention and better orchard management practices in general. The ultimate goal is to maintain the resilience and sustainability of the avocado industry in order to meet the increasing demand for this highly valued fruit on a global scale, minimise losses, and maximise yields for growers.

1.2. Main Contributions

1. **Advanced Preprocessing Techniques**: Implementing data normalization for preprocessing to enhance the quality of input data.
2. **Effective Feature Extraction with ResNet-50**: Utilizing ResNet-50, a deep convolutional neural network, for feature extraction, enabling the capture of intricate patterns and features from avocado fruit images.
3. **Innovative Classification Approach with Vision Transformer**: Introducing a novel classification approach using a vision transformer, leveraging its attention mechanism to effectively discern disease patterns and classify avocado fruit images accurately.
4. **Efficient Hyperparameter Optimization with CMBO**: Employing Cat and Mouse-Based Optimization (CMBO) to tune hyperparameters effectively, enhancing model performance and ensuring optimal results.

1.3. Organization of the Paper

This is the format for the rest of the essay. The second section will include some highlighted pertinent literature. The suggested method is explained in Section 3. Explanation of the experiments and outcomes is given in Section 4. Section 5 summarises the findings and offers some conclusions

2. Related works

In the research by Campos-Ferreira, U.E., et al. [16], Anthracnose [Colletotrichum spp.], scab [Sphaceloma perseae], and healthy fruit were the three target classes that the three learning classifiers support vector machine (SVM), random forest (RF), and multilayer perceptron (MLP) were tested to identify from digital fruit images. The RF classifier was used to compare two colour descriptor extraction methods: region selection and image subsampling. The results showed an overall classification accuracy (ACC) of 84±0.08% with subsampling and 98±0.03% with region selection. The colour descriptors that were extracted using region selection were then used to evaluate the classifiers. SVM was inferior to RF and MLP, with an ACC of 98±0.03%. Anthracnose and scab were distinguished with an F1 score of 98%.

The study by Matsui. T et al. [17] sought to determine how well a deep learning-based semantic segmentation detection model would identify Hass avocados can develop two types of fruit rot: stem-end and body rot. Consequently, 5-fold cross-validation was used to train

and validate U-net+ + in order to determine if every pixel in an X-ray picture is infected or not. Each X-ray image was then binary classified with an accuracy of 0.98 according to whether internal fruit rots were present or absent. Furthermore, a 3.15 percent RMSE was used to quantify the percentage of infected area. The suggested method identified rot along low-contrast fruit edges in addition to stem-end and body rot. The findings of this study suggest that decay inside the Hass avocado fruit could be successfully detected by the recommended automated inspection system using X-ray image analysis and deep learning.

In the study by Gulzar, Y. [18], For the experiments, a dataset comprising 26,149 photos of 40 distinct various fruit varieties were used. The test and training sets were divided into 3:1 ratios and randomly recreated. The experiment involved the introduction of a five-layer customised head into the MobileNetV2 architecture. The customised head was used to replace Layer of Classification in MobileNetV2 Model, resulting in the creation of the TL-MobileNetV2 modified version of MobileNetV2. Furthermore, the pre-trained model was retained through the use of transfer learning. TL-MobileNetV2 outperformed MobileNetV2 by 3% with an accuracy of 99%, and TL-MobileNetV2 had an identical 1% error rate. The accuracy was 8, 11, 6, and 10% higher than that of AlexNet, VGG16, InceptionV3, and ResNet, respectively. In addition, the TL-MobileNetV2 model achieved 99% F1-score, 99% recall, and 99% precision.

Pawar, S.E., et al. [19] suggested a method for classifying and detecting fruit diseases that combines deep learning and hybrid machine learning. On heterogeneous fruit datasets, a variety of feature extraction and selection strategies were used, along with machine learning and deep learning classification algorithms. The suggested hybrid CNN obtained the best accuracy of 97.10% across all fruit image datasets in a thorough experimental analysis.

The object of the study by Guo, X., et al. [20] was to create a hyperspectral imaging (HSI) method for estimating internal chilling injury in avocados. Hyperspectral cameras covering the 400 nm to 1000 nm wavelength range were used to take 2 nm resolution images of the avocado surface. Using estimated functions of multiple instances (eFUMI) and hyperspectral unmixing techniques, the spectra that correlated with internal injured areas were identified. Measuring the firmness of the fruit and the extent of injuries revealed a correlation between the external and internal alterations detected by HSI, indicating that HSI may be utilised to track avocado internal disorders.

The research by Wijaya, Y.F., et al. [21] concentrated on filling the void by utilising the created Convolutional Neural Network architecture on a large and varied dataset consisting of 67,692 image files divided into 131 fruit classes. With training accuracy reaching 98.39% and validation accuracy at 90%, the training process demonstrated a significant improvement in accuracy. Training loss also decreased to 0.0430 and validation loss to 0.2991. With a shallow loss of 0.0251 in the 59th epoch, the training accuracy reached its peak in the advanced stage of training at 99.43%.

The work by Zala, S., et al. [22] altered utilising techniques based on deep learning, particularly the use of a Vision Transformer and a customised Convolutional Neural Network (CNN). Twelve different fruit classes six categories of healthy fruits and six categories of unhealthy fruits made up the dataset used in this study. For training and testing, a total of 12,000 data points from 12 distinct classes were utilised. To improve the models' performance, a great deal of preprocessing was done along with data augmentation

techniques. These actions were taken to make sure the models could accurately represent and distinguish between the distinctive qualities and traits of different fruits. Numerous systematic research trials were carried out, entails modifying various parameters to maximise the performance of the models. The models were adjusted through this iterative process to obtain the maximum average classification accuracy. Remarkably, the vision transformer method proved to be the most effective, achieving an exceptional 98.05% average accuracy.

2.1. Research Gaps

Though deep learning techniques have made great strides in the classification of avocado fruit diseases, there are still a number of unanswered questions. First off, while previous research has concentrated on particular illnesses like scab and anthracnose, comprehensive models that can identify a wider variety of avocado diseases, including emerging pathogens, are required. Furthermore, most studies have focused on visible light images rather than exploring the possibilities of other imaging modalities like hyperspectral and X-ray imaging for disease detection. By incorporating these modalities into deep learning frameworks, the range of applications could be increased and disease detection accuracy could be improved. The difficulties of real-world deployment, such as environmental variability and scalability to large-scale agricultural operations, are also not well studied. Closing these gaps will help develop reliable, useful methods for automated management and detection of avocado fruit diseases.

3. Proposed Methodology

Figure 1 shows the proposed work flow of the Avocado fruit disease detection model using Vision Transformer.

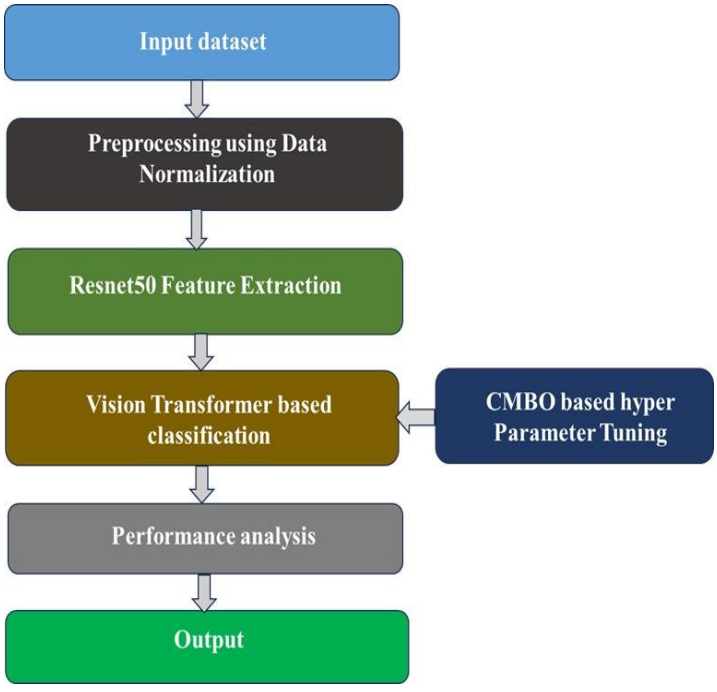

Figure 1: Block Diagram

3.1 Image Dataset

Images of avocado fruit diseases captured in real time are gathered from reliable web sources as shown in figure 2 [23]. The resulting dataset is small in comparison to what needs

in order to achieve higher accuracy a large number of images. To increase the size of the dataset, image augmentation techniques like flipping, rotating, and resizing are applied, along with colour augmentation techniques like brightness and contrast. Nine thousand photos of avocado fruits from eight different classes are included in the dataset as shown in table 1. Eighty percent of the total data is used for testing, and the remaining twenty percent is used to train the model. This divides the datasets into training and testing portions. The testing and training images' 224 x 224 pixel resizing complies with the dimensional requirement of the model. Table 2 displays how many photos were used to train and test the suggested model.

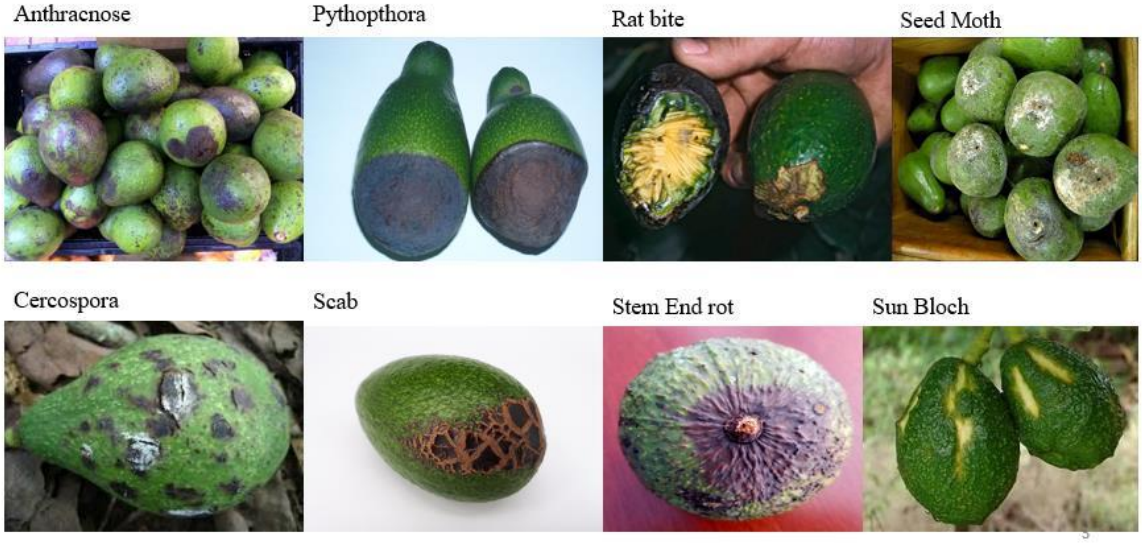

Figure 2: Examples of avocado fruit illnesses in pictures

Table.1 Specifics of the classes of diseases affecting avocado fruit

Disease Classes	**Number of images**
Cercospora	845
Rat Bite	616
Stem End rot	344
Scab	1780
Phytophthora	3088
Seed Moth	231
Sun bloch	2082
Anthracnose	3000

Table.2 Pictures used in testing and training

Disease Classes	**Number of Training images**	**Number of Testing images**
Seed Moth	185	46
Sun bloch	1666	416
Anthracnose	2400	60
Rat Bite	493	123

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

Cercospora	676	169
Stem End rot	275	69
Phytophthora	2470	618
Scab	1424	356

3.2. Data Normalization

During model training, an extensive range of eigenvalues will lead to instability because the Vision transformer model is extremely sensitive to the range of input features. The dataset is normalised to quicken the rate of convergence and enable it to detect minute differences between images. Equation (1) normalises the data for each image channel [24].

$$Z_i = \frac{x_i - \bar{x}}{\delta_x} \quad (1)$$

where x_i, δ_x and $\bar{x}$ are the mean, standard deviation, and consist of the standard deviation, mean, and sample values for the channel, respectively. After normalising the data to zero mean, all image pixels with values fall into a range of [-1,1], which is found by deducting the mean from the channel pixel values and dividing the result by their standard deviation[26].

3.3. ResNet50 Feature Extraction

Kaimimg He unveiled Resnet-50, a CNN variant, in 2015 [25]. It has fifty layers total; the convolution layers make up 48 of the layers, the first layer is the maximum pooling layer, and the last layer is the average pooling layer. A collection of similar or "residual" blocks make up the ResNet model. A block functions similarly to a stack of convolutional layers. The block's output is connected to its own input via an identifying mapping path. It used a bottleneck block of three layers that used 3×3 convolutions to lower and re-establish the channel depth in order to lessen the computational load during the 3×3 convolution calculation. It have one layer, which is a $7x7$ kernel size with 64 distinct kernels. In the second convolution, 64 kernels with a 1×1 kernel size, 64 kernels with a 3×3 kernel size, and 64 kernels with a 1×1 kernel size result in nine layers. The third convolution yields 12 layers when using 1×1 kernel size and 128 kernels, the fourth convolution uses 1×1 kernel size and 512 kernels, and the fifth convolution uses 3×3 kernel size and 128 kernels. The fourth convolution yields eighteen layers with 256 kernels for 1×1 kernel size, 256 kernels for 3×3 kernel size, and 1024 kernels for 1×1 kernel size. Five and six convolutions yield nine and one layer, accordingly, so there are a total of fifty layers. As a result, a new, more compact image data matrix is created and distributed throughout the network.

3.4. Vision Transformer Classification

Due in large part to their attention mechanisms, Transformers have demonstrated impressive performance in tasks involving natural language processing. Expanding on this idea, the Vision Transformer (ViT) has become a potent image classification architecture. Three essential components make up the ViT architecture [26].

- **Patch Embedding**

The fruit pictures $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ is separated into patches of a fixed size and then flattened 2D patches are sequentially represented as $\mathbf{x}_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$, where H stands for the height of the

image, W for the width, C for the total amount of channels, and (P, P) indicates each image patch's resolution. It is possible to calculate the amount of patches N as

$$N = \frac{H \times W}{P^2} \tag{2}$$

The patches are given a linear projection before being fed into the Transformer in sequence. In this linear projection, an embedding matrix E is multiplied by the patches to map them to a D-dimensional vector space. A patch embedding is what this linear projection's result is known as. The model can determine the positional information of the image thanks to positional embeddings $\mathbf{E}_{\text{pos}}$ to the patch embeddings as an appendix. Furthermore, a learnable class token is concatenated with the embedded image patches $\mathbf{x}_{\text{class}}$, which is necessary for the process of classification. The first embedding of the patch $\mathbf{z}_0$, comprising the class token and the embedded series of image patches, is calculated as follows:

$$\mathbf{z}_0 = \left[\mathbf{x}_{\text{class}}\,; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \cdots; \mathbf{x}_p^N\mathbf{E}\right] + \mathbf{E}_{pos},\ \mathbf{E} \in \mathbb{R}^{\left(P^2 \cdot C\right) \times D}, \mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D} \tag{3}$$

Here, $\mathbf{x}_p^n$ denotes the image patch with n values between 1 and N. The Transformer encoder receives the embedded image patches that result.

- **Transformer Encoder**

Two layers of feed-forward multi-layer perceptrons (MLPs) with full connectivity and multi-head self-attention (MSA) layers make up each of the L identical encoder blocks that make up the Transformer encoder. The sequence of input from the preceding layer is received by each encoder block's l-th layer $\mathbf{z}_{\ell-1}$. The input $\mathbf{z}_{\ell-1}$ receives layer normalisation, which enhances training speed and performance by normalising throughout the feature dimension, the input values. After that, the layer normalisation output is sent to the MSA layer.

After that, the MSA layer's output is once more layer-normalized. Ultimately, the MLP layer receives the outcomes of the layer normalisation. The encoder block uses residual connections, sometimes referred to as skip connections, to help information move between non-adjacent layers. The problem of vanishing gradients is resolved by these connections, which enable gradients to move throughout the network unaffected by non-linear activation functions. In the l-th encoder layer, the gradient flow is described as

$$\mathbf{z}'_\ell = \text{MSA}\left(\text{LN}\left(\mathbf{z}_{\ell-1}\right)\right) + \mathbf{z}_{\ell-1},\ \ell = 1, \ldots, L \tag{4}$$

$$\mathbf{z}_\ell = \text{MLP}\left(\text{LN}\left(\mathbf{z}'_\ell\right)\right) + \mathbf{z}'_\ell,\ \ell = 1, \ldots, L \tag{5}$$

where layer normalisation is represented by LN.
The MSA consists of a final linear layer, a concatenation layer, a linear layer, and a self-attention layer. The number of heads k determines how many self-attention operations are carried out in parallel in the MSA. Three weight matrices are multiplied by the patch embedding in D dimensions z in each head $\mathbf{U}_q, \mathbf{U}_k$, and $\mathbf{U}_v$ to acquire the query $(\mathbf{q}), key(\mathbf{k})$, and value (v) matrices. Each head's multiplication process is described as

$$[\mathbf{q}, \mathbf{k}, \mathbf{v}] = \left[\mathbf{z}\mathbf{U}_q, \mathbf{z}\mathbf{U}_k, \mathbf{z}\mathbf{U}_v\right],\ \mathbf{U}_q, \mathbf{U}_k, \mathbf{U}_v \in \mathbb{R}^{D \times D_h} \tag{6}$$

The weighted sum of all values V is found following the projection, which is initiated after the matrices q, k, and v are obtained and divided into k subspaces. Based on the (i, j) dot product of each pair of elements, attention weights in each head are calculated their relationship as $\mathbf{q}^i$ and $\mathbf{k}^j$. The sequence's significance for each patch is indicated by the dot

product that results. Determine the weights based on the values by computing q + k as a dot product and using the softmax function, as shown below:

$$A = \text{softmax}\left(\frac{\mathbf{q}\mathbf{k}^{\top}}{\sqrt{D_h}}\right),\ A \in \mathbb{R}^{N\times N} \tag{7}$$

where $D_h = \frac{D}{k}$.

The dot product that is produced afterwards shows how significant each patch is in the order. Ascertain the weights on the values by computing q + k as the dot product and applying the softmax function, as indicated below $\mathbf{U}_{msa}$, resulting in

$$\text{MSA}\,(\mathbf{z}) = [\text{SA}_1(z); \text{SA}_2(z); \cdots; \text{SA}_k(z)]\mathbf{U}_{msa},\ \mathbf{U}_{msa} \in \mathbb{R}^{k\cdot D_h\times D} \tag{8}$$

Since each head of the MSA gathers data from various angles and locations, the model can encode complex features concurrently.

- **Classification**

The Gaussian error linear unit (GeLU) serves as the basis for the activation function of the two fully connected layers of the multi-layer perceptron (MLP), classifies the ViT model. The inputs are given a weight by the GeLU activation function based more on their values than their indicators. In contrast to the ReLU function, the GeLU function has a larger degree of curvature and can result in both favourable and unfavourable outcomes. When compared to the ReLU function, this characteristic enables the GeLU function for more accurate complex function approximation.

The final layer of the encoder chooses the sequence's first token, z_L^0, and uses layer normalisation to create the image representation r. For classification, the resultant r is then run through a tiny MLP head with a sigmoid function and one hidden layer. This is how to get the sequence's image representation:

$$\mathbf{r} = \text{LN}\,(\mathbf{z}_L^0) \tag{9}$$

3.5. Hyper parameter tuning using CMBO

The mathematical model for the Cat and Mouse-Based Optimisation Algorithm (CMBO) is presented in this section along with an explanation of its theory for use in tuning hyper parameters [27].

The population-based algorithm known as the CMBO was inspired by the way a cat would naturally attack a mouse and then flee to a safe haven. Two packs of mice and cats are the search agents in the suggested algorithm, and they move randomly throughout the problem search space. The proposed algorithm updates the population members in two stages. The first phase models how cats approach mice, and the second phase models how mice flee to safe havens in order to escape and survive.

Every person in the population is, mathematically speaking, a suggested solution to the issue. As a matter of fact, A member of the population specifies values for the problem variables by using their location in the search space. Each individual in the population is therefore a vector, and the variables of the problem are determined by their values. Equation (10) uses a matrix known as the population matrix to determine the population of the algorithm.

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & . & \vdots \\ x_{i,1} & \cdots & x_{i,d} & \cdots & x_{i,m} \\ \vdots & . & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,d} & \cdots & x_{N,m} \end{bmatrix}_{N\times m} \quad (10)$$

wherein X is the CMBO population matrix, X_i is the i th search agent, $x_{i,d}$ is the value that the ith search agent obtained for the dth problem variable, N is the amount of members in the population, and m is the amount of problem variables.

As previously stated, the recommended values for the variables in question are decided by each individual in the population. As a result, the objective function has a value assigned to each member of the population. In Equation (11) a vector represents the values acquired for the objective function.

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} \quad (11)$$

wherein F is the vector of values for the objective function and F_i represents the value of the ith search agent's objective function.

Based on values found for the objective functions, the population members are ranked from best to worst, with the lowest objective function value and the highest objective function value. The sorted population matrix and the sorted objective function are found using equations (12) and (13).

$$X^S = \begin{bmatrix} X_1^S \\ \vdots \\ X_i^S \\ \vdots \\ X_N^S \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1}^S & \cdots & x_{1,d}^S & \cdots & x_{1,m}^S \\ \vdots & \ddots & \vdots & . & \vdots \\ x_{i,1}^S & \cdots & x_{i,d}^S & \cdots & x_{i,m}^S \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ x_{N,1}^S & \cdots & x_{N,d}^S & \cdots & x_{N,m}^S \end{bmatrix}_{N\times m} \quad (12)$$

$$F^S = \begin{bmatrix} F_1^S & \min(F) \\ \vdots & \vdots \\ F_N^S & \max(F) \end{bmatrix}_{N\times 1} \quad (13)$$

where X^S is the population matrix that has been sorted using the objective function value, X_i^S is the i th a component of the sorted population matrix, $x_{i,d}^s$ is the ith search agent's value for the dth problem variable in the sorted population matrix, and F^S is an objective function's sorted vector.

The two mouse and cat groups that comprise the population matrix are part of the proposed CMBO. The other half of the members, who gave lower values for the same objective function, make up the population of cats, according to the CMBO, which also defines the population of mice. Higher values for the goal function were supplied by the first half of the members. Equations (14) and (15) are used, respectively, to determine the populations of mice and cats based on this idea.

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

$$M = \begin{bmatrix} M_1 = X_1^S \\ \vdots \\ M_i = X_i^S \\ \vdots \\ M_{N_m} = X_{N_m}^S \end{bmatrix}_{N_m \times m} = \begin{bmatrix} x_{1,1}^S & \cdots & x_{1,d}^S & \cdots & x_{1,m}^S \\ \vdots & \ddots & \vdots & . & \vdots \\ x_{i,1}^S & \cdots & x_{i,d}^S & \cdots & x_{i,m}^S \\ \vdots & \therefore & \vdots & \ddots & \vdots \\ x_{N_m,1}^S & \cdots & x_{N_m,d}^S & \cdots & x_{N_m,m}^S \end{bmatrix}_{N_m \times m} \quad (14)$$

$$C = \begin{bmatrix} C_1 = X_{N_m+1}^S \\ \vdots \\ C_j = X_{N_m+j}^S \\ \vdots \\ C_{N_c} = X_{N_m+N_c}^S \end{bmatrix}_{N_c \times m} = \begin{bmatrix} x_{N_m+1,1}^S & \cdots & x_{N_m+1,d}^S & \cdots & x_{N_m+1,m}^S \\ \vdots & \ddots & \vdots & . & \vdots \\ x_{N_m+j,1}^S & \cdots & x_{N_m+j,d}^S & \cdots & x_{N_m+j,m}^S \\ \vdots & \therefore & \vdots & \ddots & \vdots \\ x_{N_m+N_c,1}^S & \cdots & x_{N_m+N_c,d}^S & \cdots & x_{N_m+N_c,m}^S \end{bmatrix}_{N_c \times m} \quad (15)$$

where M is the mouse population matrix, N_m is the quantity of mice, M_i is the j th mouse, C is the cat population matrix, N_c is the quantity of felines, and C_j is the cat that is i th.

In the initial stage, cats' behaviour towards mice and their natural behaviour are used to model how their positions change updating the search parameters. Equations (16) through (18) are used to mathematically model this stage of the proposed CMBO update.

$$C_j^{\text{new}} : c_{j,d}^{\text{new}} = c_{j,d} + r \times \left(m_{k,d} - I \times c_{j,d}\right) j = 1 : N_c, d = 1 : m, k \in 1 : N_m \quad (16)$$

$$I = \text{round}\ (1 + \ \text{rand}\ ) \quad (17)$$

$$C_j = \begin{cases} C_j^{\text{new}}, \mid F_j^{c,\ \text{new}} < F_j^c \\ C_j, \mid \ \text{else} \end{cases} \quad (18)$$

Here, C_j^{new} is the J th cat's current status, $c_{j,d}^{\text{new}}$ is the new value that the j-th cat found for the d −th problem variable, where r is a random number between 0 and 1, $m_{k,d}$ is the k th mouse's d th dimension, $F_j^{c,\ \text{new}}$ is the goal function value derived from the updated status of the jth cat.

The suggested CMBO models' second section mice's escape to havens. According to CMBO, each mouse has a randomly assigned haven, and the mice seek shelter in these havens. The search space's havens' locations can be determined by patterning the locations of different algorithm members are generated at random. Equations (19)–(21) are used to mathematically model this phase of the mice's position updates.

$$H_i : h_{i,d} = x_{l,d} \& i = 1 : N_m, d = 1 : m, l \in 1 : N \quad (19)$$

$$M_i^{\text{new}} : m_{i,d}^{\text{new}} = m_{i,d} + r \times \left(h_{i,d} - I \times m_{i,d}\right) \times \text{sign}\ (F_i^m - F_i^H)\ i = 1 : N_m, d = 1 : m \quad (20)$$

$$M_i = \begin{cases} M_i^{\text{new}}, \mid F_i^{m,\ \text{new}} < F_i^m \\ M_i, \mid \ \text{else} \end{cases} \quad (21)$$

Herein, H_i is the haven for the i th mouse and F_i^H is the value of its objective function. M_i^{new} is the mouse's current state, and $F_i^{m,new}$ is the value of its objective function.

The algorithm moves on to the next iteration once every member of the population has been updated. According to Equations (14)–(21), this iteration of the algorithm keeps running until

the specified stop condition is satisfied. An optimisation algorithm may terminate after a set number of iterations or by specifying an acceptable degree of error between the answers found in ensuing iterations. Furthermore, the algorithm may be stopped after a predetermined amount of time. The best yielded quasi-optimal solution is obtained by using the CMBO after the iterations are finished and the algorithm is fully applied to the optimisation problem. Algorithm 1 also presents flowcharts of the different phases of the suggested CMBO, each of which has pseudocode specifications.

Algorithm 1 Pseudocode of CMBO

$Start\ CMBO.$

$Input\ problem\ information: variables, objective\ function, and\ constraints.$

$Set\ number\ of\ search\ agents\ (N)\ and\ iterations\ (T).$

$Generate\ an\ initial\ population\ matrix\ at\ random.$

$Evaluate\ the\ objective\ function.$

$For\ t = 1:T$

$Sort\ population\ matrix\ based\ on\ objective\ function\ value\ using\ Equations\ (12)\ and\ (13).$

$Select\ population\ of\ mice\ M\ using\ Equation\ (14).$

$Select\ population\ of\ cats\ C\ using\ Equation\ (15).$

$Phase\ 1: update\ status\ of\ cats.$

$For\ j = 1:N_c$

$Update\ status\ of\ the\ j\ th\ cat\ using\ Equations\ (16) - (18).$

End

$Phase\ 2: update\ status\ of\ mice.$

$For\ i = 1:N_m$

$Create\ haven\ for\ the\ i\ th\ mouse\ using\ Equation\ (19).$

$Update\ status\ of\ the\ i\ th\ mouse\ using\ Equations\ (20)\ and\ (21).$

end
end

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

4. Results and Discussion

4.1. Experimental Setup

Using Google Colab, the experiment is run on an Intel Core i5-8200 CPU running at 2.60 GHz with 8GB of RAM and an AMD Radeon M430S graphics card. Python programming and deep learning frameworks like TensorFlow and Keras are used to develop the model.

4.2. Performance Evaluation

The proposed model's performance is displayed in an error-specific table called the confusion matrix or error matrix. For binary classification, it can be a 2x2 table; for multiway classification, it can be a table with additional dimensions. Our approach predicts whether a fruit is diseased or not by using a binary classifier. It has two rows and two columns with the following values in them.

Normal- Absence of diseases; Abnormal- Presence of disease

True positive (TP)- No of samples that were correctly classified as abnormal

False positive (FP)- No of samples that were incorrectly classified as abnormal

True Negative (TN)- No of samples that were correctly classified as normal

False Negative (FN)- No of samples that were incorrectly classified as normal.

Specificity: It is shown as the percentage of the real negative that was predicted to be negative. It is represented as

$$\text{Specificity (SP)} = \frac{TN}{TN+FP} \tag{22}$$

Recall: It shows the percentage of real positives that are accurately identified. It can be ascertained by

$$\text{Recall (RC)} = \frac{TP}{TP+FN} \tag{23}$$

F-score: It is a gauge of the model's precision using the given dataset. It is comparable to what follows.

$$F - \text{ score (F1)} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{24}$$

Precision: Its definition is the percentage of correctly predicted positives that turn out to be real positives. It is stated aa:

$$\text{Precision (PR)} = \frac{\text{TN}}{\text{TN+FP}} \tag{25}$$

Accuracy: When comparing the quantity of observations that were accurately predicted to the total number of observations, the most institutional performance is measured. It describes the capacity to discern between typical and unusual situations. It is stated as:

$$\text{Accuracy (ACC)} = \frac{TP+TN}{TP+TN+FP+FN} \tag{26}$$

Type-I and type-II errors are represented by false positives and false negatives, respectively.

4.3. Various types of diseases classification

Table 3 illustrates the various types of avocado fruit disease classification achieved using the proposed Vision Transformer (ViT) model. The classification results showcase the model's efficacy in accurately identifying different diseases affecting avocado fruits.

Table 3: Different types of Avocado fruit disease classification using proposed ViT model.

Disease Classes	ACC	PR	RC	F1	SP
Anthracnose	99.42	99.71	99.94	99.38	99.22
Cercospora	99.16	99.36	98.89	99.96	99.75
Phytophthora	99.84	98.98	99.78	99.91	99.53
Rat Bite	99.25	98.81	99.42	99.88	99.44
Scab	99.39	99.93	99.66	99.12	99.51
Seed Moth	98.90	99.31	99.75	99.25	99.34
Stem End rot	99.71	99.87	99.86	99.88	99.66
Sun bloch	99.58	99.43	99.61	99.32	99.14

Table 3 and figure 3 summarizes the suggested classification performance Vision Transformer (ViT) model across various types of avocado fruit diseases. The ViT model demonstrates high ACC in distinguishing between healthy fruits and those affected by different diseases, achieving an ACC of 99.42% for Anthracnose, 99.16% for Cercospora, 99.84% for Phytophthora, 99.25% for Rat Bite, 99.39% for Scab, 98.90% for Seed Moth, 99.71% for Stem End rot, and 99.58% for Sun bloch. PR values range from 98.81% to 99.93%, demonstrating the model's capacity to identify positive instances with ACC. Similarly, RC values, representing the model's sensitivity, range from 98.89% to 99.94%, indicating its effectiveness in identifying true positives. The F1, which illustrates the model's overall performance in disease classification, ranges from 99.12% to 99.96% and balances PR and RC. Furthermore, SP values, which indicate how well the model can classify negative instances, range from 99.14% to 99.75%, further affirming the robustness of the ViT model in distinguishing between healthy and diseased avocado fruits across various disease classes.

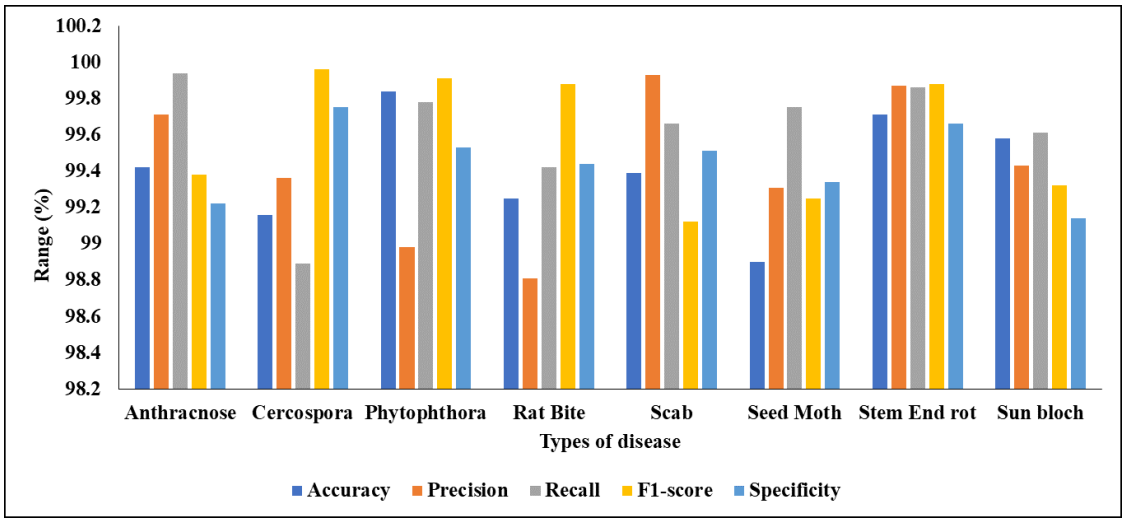

Figure 3: Graphical validation of types of disease

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

4.4. Classification validation

Table 4 presents the classification analysis conducted using a various existing models. The table provides a comparative overview of the ACC, PR, RC, and F1 metrics achieved by each model, facilitating a comprehensive evaluation of their effectiveness in avocado fruit disease classification.

Table 4: Classification analysis using various existing models

Models	ACC (%)	PR (%)	RC (%)	F1 (%)
RNN	91.44	90.33	90.21	90.28
AlexNet	81.55	80.45	79.86	79.78
ResNet	97.62	96.52	96.44	96.15
DenseNet	98.47	98.37	98.26	98.18
Proposed ViT model	99.98	99.39	99.29	99.27

Table 4 and figure 4 presents a comparative analysis of classification performance using various existing models for avocado fruit disease detection. The Recurrent Neural Network (RNN) achieved an ACC of 91.44%, with PR, RC, and F1 scores of 90.33%, 90.21%, and 90.28%, respectively. AlexNet exhibited an ACC of 81.55%, with PR, RC, and F1 scores of 80.45%, 79.86%, and 79.78%, respectively. ResNet demonstrated improved performance with an ACC of 97.62%, PR of 96.52%, RC of 96.44%, and F1 score of 96.15%. DenseNet further enhanced classification ACC to 98.47%, with PR, RC, and F1 scores of 98.37%, 98.26%, and 98.18%, respectively. Notably, the proposed Vision Transformer (ViT) model outperformed all other models, achieving an impressive ACC of 99.98%, PR of 99.39%, RC of 99.29%, and F1 score of 99.27%. These results underscore the superior performance of the ViT model in avocado fruit disease classification compared to conventional models like RNN, AlexNet, ResNet, and DenseNet.

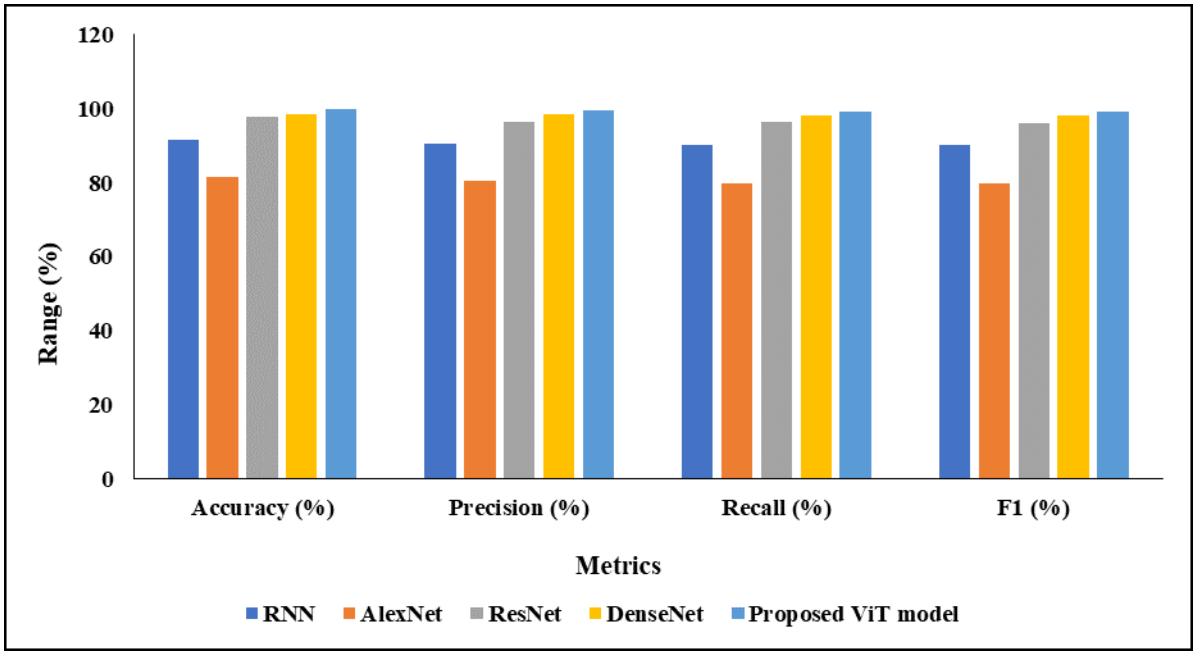

Figure 4: Graphical validation of Classification analysis

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

4.5. Training and testing validation

Table 5 depicts the comprehensive ACC analysis of training and testing validation conducted as part of this study.

Table 5: The ACC analysis of training and testing validation

Models	60:40	70:30	80:20
RNN	91.61	92.14	93.16
AlexNet	91.53	92.25	93.48
ResNet	92.34	93.58	94.34
DenseNet	93.65	94.69	95.66
Proposed ViT model	96.89	97.88	99.98

Table 5 and figure 5 illustrates the ACC analysis of training and testing validation for various models across different train-test split ratios. The Recurrent Neural Network (RNN) achieved accuracies of 91.61%, 92.14%, and 93.16% for the 60:40, 70:30, and 80:20 split ratios, respectively. Similarly, AlexNet demonstrated accuracies of 91.53%, 92.25%, and 93.48% across the same split ratios. ResNet exhibited increasing accuracies of 92.34%, 93.58%, and 94.34% as the train-test split ratio widened. DenseNet showed consistent improvement in ACC, achieving 93.65%, 94.69%, and 95.66% ACC across the three split ratios. Notably, the proposed Vision Transformer (ViT) model consistently outperformed other models, with accuracies of 96.89%, 97.88%, and an exceptional 99.98% for the 60:40, 70:30, and 80:20 split ratios, respectively. These results highlight the superior performance and scalability of the ViT model across different training and testing data distributions compared to traditional models like RNN, AlexNet, ResNet, and DenseNet.

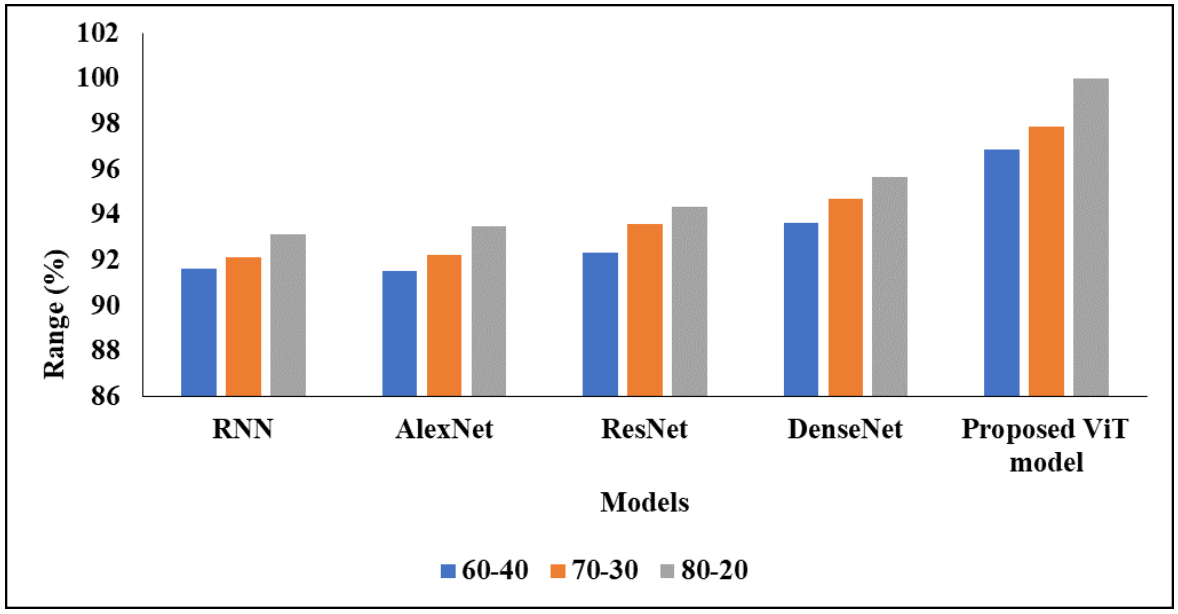

Figure 5: Graphical validation of testing and training phase

5. Conclusion

The study concludes by presenting a thorough framework for classifying avocado fruit diseases using cutting-edge deep learning methods. Our suggested approach has proven to be effective in addressing the difficulties related to disease detection in avocado

cultivation, as evidenced by thorough testing and assessment. We have accomplished remarkable accuracy in disease classification by using sophisticated preprocessing techniques for data normalisation, utilising ResNet-50 for feature extraction, and utilising a vision transformer for classification. Moreover, the optimal performance of our model has been guaranteed by our application of Cat and Mouse-Based Optimisation (CMBO) for hyperparameter tuning. Our approach is robust and reliable, as evidenced by its superior classification accuracy of 99.98%, which surpasses current models and sets a new benchmark in the classification of avocado fruit diseases. The study has important ramifications for the sustainability and adaptability of the avocado sector in addition to advancing automated orchard management techniques. Further development and implementation of our suggested framework could transform disease control tactics in the future, increasing avocado yield and guaranteeing food security for people throughout the world.

References:

[1]. Thangaraj, R., Dinesh, D., Hariharan, S., Rajendar, S., Gokul, D., & Hariskarthi, T. R. (2020). Automatic recognition of avocado fruit diseases using modified deep convolutional neural network. *International Journal of Grid and Distributed Computing*, *13*(1), 1550-1559.

[2]. Campos-Ferreira, U. E., González-Camacho, J. M., & Carrillo-Salazar, A. (2023). Automatic identification of avocado fruit diseases based on machine learning and chromatic descriptors. *Revista Chapingo. Serie horticultura*, *29*(3), 115-130.

[3]. Jaramillo-Acevedo, C. A., Choque-Valderrama, W. E., Guerrero-Álvarez, G. E., & Meneses-Escobar, C. A. (2020). Hass avocado ripeness classification by mobile devices using digital image processing and ANN methods. *International Journal of Food Engineering*, *16*(12), 20190161.

[4]. Abdulridha, J., Ehsani, R., Abd-Elrahman, A., & Ampatzidis, Y. (2019). A remote sensing technique for detecting laurel wilt disease in avocado in presence of other biotic and abiotic stresses. *Computers and electronics in agriculture*, *156*, 549-557.

[5]. Mejiá-Cabrera, H. I., Flores, J. N., Sigueñas, J., Tuesta-Monteza, V., & Forero, M. G. (2020, August). Identification of Lasiodiplodia Theobromae in avocado trees through image processing and machine learning. In *Applications of Digital Image Processing XLIII* (Vol. 11510, pp. 572-580). SPIE.

[6]. Salazar-Reque, I. F., Pacheco, A. G., Rodriguez, R. Y., Lezama, J. G., & Huamán, S. G. (2019, April). An image processing method to automatically identify Avocado leaf state. In *2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA)* (pp. 1-5). IEEE.

[7]. Bautista-Baños, S., Ventura-Aguilar, R. I., & de Lorena Ramos-García, M. (2019). Avocado. In *Postharvest pathology of fresh horticultural produce* (pp. 227-256). CRC Press.

[8]. Everett, K. R. (2020). Avocado diseases affecting fruit quality. *CABI Reviews*, (2020).

[9]. Valiente, L. D., Parco, K. M. R., & Sangalang, G. C. P. (2021, October). Non-destructive image processing analysis for defect identification and maturity detection on avocado fruit.

In *2021 5th International Conference on Communication and Information Systems (ICCIS)* (pp. 175-179). IEEE.

[10]. Fuentes-Aragón, D., Silva-Rojas, H. V., Guarnaccia, V., Mora-Aguilera, J. A., Aranda-Ocampo, S., Bautista-Martínez, N., & Téliz-Ortíz, D. (2020). Colletotrichum species causing anthracnose on avocado fruit in Mexico: Current status. *Plant Pathology*, *69*(8), 1513-1528.

[11]. Nasir, I. M., Bibi, A., Shah, J. H., Khan, M. A., Sharif, M., Iqbal, K., ... & Kadry, S. (2021). Deep learning-based classification of fruit diseases: An application for precision agriculture. *Comput. Mater. Contin*, *66*(2), 1949-1962.

[12]. Khan, M. A., Akram, T., Sharif, M., & Saba, T. (2020). Fruits diseases classification: exploiting a hierarchical framework for deep features fusion and selection. *Multimedia Tools and Applications*, *79*, 25763-25783.

[13]. Agarwal, A., Sarkar, A., & Dubey, A. K. (2019). Computer vision-based fruit disease detection and classification. In *Smart Innovations in Communication and Computational Sciences: Proceedings of ICSICCS-2018* (pp. 105-115). Springer Singapore.

[14]. Nikhitha, M., Sri, S. R., & Maheswari, B. U. (2019, June). Fruit recognition and grade of disease detection using inception v3 model. In *2019 3rd International conference on electronics, communication and aerospace technology (ICECA)* (pp. 1040-1043). IEEE.

[15]. Ayyub, S. R. N. M., & Manjramkar, A. (2019, March). Fruit disease classification and identification using image processing. In *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 754-758). IEEE.

[16]. Campos-Ferreira, U. E., González-Camacho, J. M., & Carrillo-Salazar, A. (2023). Automatic identification of avocado fruit diseases based on machine learning and chromatic descriptors. Revista Chapingo. Serie horticultura, 29(3), 115-130.

[17]. Matsui, T., Sugimori, H., Koseki, S., & Koyama, K. (2023). Automated detection of internal fruit rot in Hass avocado via deep learning-based semantic segmentation of X-ray images. Postharvest Biology and Technology, 203, 112390.

[18]. Gulzar, Y. (2023). Fruit image classification model based on MobileNetV2 with deep transfer learning technique. Sustainability, 15(3), 1906.

[19]. Pawar, S. E., Surana, A. V., Sharma, P., & Pujeri, R. (2024). Fruit Disease Detection and Classification using Machine Learning and Deep Learning Techniques. International Journal of Intelligent Systems and Applications in Engineering, 12(4s), 440-453.

[20]. Guo, X., Tseung, C., Zare, A., & Liu, T. (2023). Hyperspectral image analysis for the evaluation of chilling injury in avocado fruit during cold storage. Postharvest Biology and Technology, 206, 112548.

[21]. Wijaya, Y. F., & Hindarto, D. (2024). Advancing Fruit Image Classification with State-of-the-Art Deep Learning Techniques. Sinkron: jurnal dan penelitian teknik informatika, 8(2), 1125-1134

[22]. Zala, S., Goyal, V., Sharma, S., & Shukla, A. (2024). Transformer based fruits disease classification. Multimedia Tools and Applications, 1-21.

[23]. Thangaraj, R., Dinesh, D., Hariharan, S., Rajendar, S., Gokul, D., & Hariskarthi, T. R. (2020). Automatic recognition of avocado fruit diseases using modified deep convolutional neural network. *International Journal of Grid and Distributed Computing*, *13*(1), 1550-1559.

[24]. Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, *97*, 105524.

[25]. Li, B., & Lima, D. (2021). Facial expression recognition via ResNet-50. *International Journal of Cognitive Computing in Engineering*, *2*, 57-64.

[26]. Lee, C. P., Lim, K. M., Song, Y. X., & Alqahtani, A. (2023). Plant-CNN-ViT: plant classification with ensemble of convolutional neural networks and vision transformer. *Plants*, *12*(14), 2642.

[27]. Dehghani, M., Hubálovský, Š., & Trojovský, P. (2021). Cat and mouse based optimizer: A new nature-inspired optimization algorithm. *Sensors*, *21*(15), 5214.